

The RSA scheme capitalizes on the relative ease of creating a composite number from the product of two prime numbers whereas the attempt to factor the composite number into its constituent primes is difficult. The RSA scheme uses a public key E comprising a pair of positive integers n and e, where n is a composite number of the form

$$n=p \cdot q \quad (1)$$

where p and q are different prime numbers, and e is a number relatively prime to (p-1) and (q-1); that is, e is relatively prime to (p-1) or (q-1) if e has no factors in common with either of them. Importantly, the sender has access to n and e, but not to p and q. The message M is a number representative of a message to be transmitted wherein

$$0 \leq M \leq n-1. \quad (2)$$

The sender enciphers M to create ciphertext C by computing the exponential

$$[C=M^e \pmod{n}] \quad C \equiv M^e \pmod{n}. \quad (3)$$

Replace the paragraph beginning at col. 2, line 19 with the following:

The recipient of the ciphertext C retrieves the message M using a (private) decoding key D, comprising a pair of positive integers d and n, employing the relation

$$[M=C^d \pmod{n}] \quad C \equiv M^d \pmod{n} \quad (4)$$

As used in (4), above, d is a multiplicative inverse of

$$e \pmod{\text{lcm}((p-1), (q-1)))} \quad (5)$$

so that

$$[e \cdot d = 1 \pmod{\text{lcm}((p-1), (q-1)))}] \quad e \cdot d \equiv 1 \pmod{\text{lcm}((p-1), (q-1)))} \quad (6)$$

where $\text{lcm}((p-1), (q-1))$ is the least common multiple of numbers p-1 and q-1. Most commercial implementations of RSA employ a different, although equivalent, relationship for obtaining d:

$$[d = e^{-1} \pmod{(p-1)(q-1)}] \quad d \equiv e^{-1} \pmod{(p-1)(q-1)}. \quad (7)$$

This alternate relationship simplifies computer processing.

Replace the paragraph beginning at col. 3, line 23 with the following:

It is still another object of this invention to provide a system and method for implementing an RSA scheme in which the [components] factors of n do not increase in length as n increases in length.

Replace the paragraph beginning at col. 3, line 27 with the following:

It is still another object to provide a system and method for utilizing multiple (more than two), distinct prime number [components] factors to create n.

Replace the paragraph beginning at col. 3, line 36 with the following:

The present invention discloses a method and apparatus for increasing the computational speed of RSA and related public key schemes by focusing on a neglected area of computation inefficiency. Instead of $n=p \cdot q$, as is universal in the prior art, the present invention discloses a method and apparatus wherein n is developed from three or more distinct random prime numbers; i.e., $n=p_1 \cdot p_2 \cdot \dots \cdot p_k$, where k is an integer greater than 2 and p_1, p_2, \dots, p_k are sufficiently large distinct random primes. Preferably, "sufficiently large primes" are prime numbers that are numbers approximately 150 digits long or larger. The advantages of the invention over the prior art should be immediately apparent to those skilled in this art. If, as in the prior art, p and q are each on the order of, say, 150 digits long, then n will be on the order of 300 digits long. However, three primes p_1, p_2 and p_3 employed in accordance with the present invention can each be on the order of 100 digits long and still result in n being 300 digits long. Finding and verifying 3 distinct primes, each 100 digits long, requires significantly fewer computational cycles than finding and verifying 2 primes each 150 digits long.

Replace the paragraph beginning at col. 3, line 56 with the following:

The commercial need for longer and longer primes shows no evidence of slowing; already there are projected requirements for n of about 600 digits long to forestall incremental improvements in factoring techniques and the ever faster computers available to break ciphertext. The invention, allowing 4 primes each about 150 digits long to obtain a 600 digit n , instead of two primes about [350] 300 digits long, results in a marked improvement in computer performance. For, not only are primes that are 150 digits in size easier to find and verify than ones on the order of [350] 300 digits, but by applying techniques the inventors derive from the Chinese Remainder Theorem (CRT), public key cryptography calculations for encryption and decryption are completed much faster--even if performed serially on a single processor system. However, the inventors' techniques are particularly adapted to [be] advantageously apply [enable] RSA public key cryptographic operations to parallel computer processing.

Replace the paragraph beginning at col. 4, line 6 with the following:

The present invention is capable of [using] extending the RSA scheme to perform encryption and decryption operation using a large (many digit) n much faster than heretofore possible. Other advantages of the invention include its employment for decryption without the need to revise the RSA public key encryption transformation scheme currently in use on thousands of large and small computers.

Replace the paragraph beginning at col. 4, line 13 with the following:

A key assumption of the present invention is that n , composed of 3 or more sufficiently large distinct prime numbers, is no easier (or not very much easier) to factor than the prior art, two prime number n . The assumption is based on the observation that there is no indication in the prior art literature that it is "easy" to factor a product consisting of more than two sufficiently large, distinct prime numbers. This assumption may be justified given the continued effort (and failure) among experts to find a way "easily" to break large [component] composite numbers into their large prime factors. This assumption is similar, in the inventors' view, to the assumption underlying the entire

field of public key cryptography that factoring composite numbers made up of two distinct primes is not "easy." That is, the entire field of public key cryptography is based not on mathematical proof, but on the assumption that the empirical evidence of failed sustained efforts to find a way systematically to solve NP problems in polynomial time indicates that these problems truly are "difficult."

Replace the paragraph beginning at col. 4, line 32 with the following:

The invention is preferably implemented in a system that employs parallel operations to perform the encryption, decryption operations required by the RSA scheme. Thus, there is also disclosed a cryptosystem that includes a central processor unit (CPU) coupled to a number of exponentiator elements. The exponentiator elements are special purpose arithmetic units designed and structured to be provided message data M, an encryption key e, and a number n (where $[n=p_1 * p_2 * \dots * p_k]$ $n=p_1 p_2 \dots p_k$, k being greater than 2) and return ciphertext C according to the relationship,

$$[C=M^e \pmod{n}] \quad C \equiv M^e \pmod{n}$$

Replace the paragraph beginning at col. 4, line 45 with the following:

Alternatively, the exponentiator elements may be provided the ciphertext C, a decryption (private) key d and n to return M according to the relationship,

$$[M=C^d \pmod{n}] \quad M \equiv C^d \pmod{n}$$

Replace the paragraph beginning at col. 4, line 50 with the following:

According to this decryption aspect of the invention, the CPU receives a task, such as the requirement to decrypt [ciphertext] ciphertext data C. The CPU will also be provided, or have available, a [public] private key [e] d and n, and the factors of n (p_1, p_2, \dots, p_k). The CPU breaks the [encryption] decryption task down into a number of sub-tasks, and delivers the sub-tasks to the exponentiator elements. [When the] The results of the sub-tasks are returned by the exponentiator elements to the CPU which [will], using a

form of the CRT, combines the results to obtain the message data M. An encryption task may be performed essentially in the same manner by the CPU and its use of the exponentiator elements. However, usually the factors of n are not available to the sender (encryptor), only the public key, e and n, so that no sub-tasks are created.

A12
Before the paragraph beginning at col. 5, line 52, insert the following paragraph:

Alternatively, a message data M can be encoded with the private key to a signed message data M_s using a relationship of the form

$$\underline{M_s \equiv M^d \pmod{n}}.$$

A13
The message data M can be reproduced from the signed message data M_s by decoding the signed data with the public key, using a relationship of the form

$$\underline{M \equiv M_s^e \pmod{n}}.$$

A14
Replace the paragraph beginning at col. 5, line 30 with the following:

According to the present invention, the public key portion e is picked. Then, three or more random large, distinct prime numbers, p₁, p₂, . . . , p_k are developed and checked to ensure that each (p_i-1) is relatively prime to e. Preferably, the prime numbers are of equal length. Then, the product [n=p₁, p₂, . . . , p_k] n=p₁·p₂·. . . ·p_k is computed.

A15
Replace the paragraph beginning at col. 5, line 36 with the following:

Finally, the decryption [key] exponent, d, is established by the relationship:

[d=e⁻¹ mod ((p₁-1)(p₂-1) . . . (p_k-1))] d≡ e⁻¹ mod ((p₁-1)(p₂-1)· . . . ·(p_k-1)). or equivalently

$$\underline{d \equiv e^{-1} \pmod{\text{lcm}((p_1-1), (p_2-1), \dots, (p_k-1))}}$$

Replace the paragraph beginning at col. 5, line 41 with the following:

The message data, M is encrypted to ciphertext C using the relationship of (3), above, i.e.,

$$[C=M^e \bmod n] \underline{C \equiv M^e \pmod n}$$

Replace the paragraph beginning at col. 5, line 46 with the following:

To decrypt the ciphertext, C, the relationship of [(3)] (4), above, is used:

$$[M=C^d \bmod n] \underline{M \equiv C^d \pmod n}$$

where n and d are those values identified above.

Replace the paragraph beginning at col. 5, line 52 with the following:

Using the present invention involving three primes to develop the product n, RSA encryption and decryption time can be substantially less than an RSA scheme using two primes by dividing the encryption or decryption task into sub-tasks, one sub-task for each distinct prime. (However, breaking the encryption or decryption into subtasks requires knowledge of the factors of n. This knowledge is not usually available to anyone except the owner of the key, so the encryption process can be accelerated only in special cases, such as encryption for local storage. A system encrypting data for another user performs the encryption process according to (3), independent of the number of factors of n. Decryption, on the other hand, is performed by the owner of a key, so the factors of n are generally known and can be used to accelerate the process.) For example, assume that three distinct primes, p_1 , p_2 , and p_3 , are used to develop the product n. Thus, decryption of the ciphertext, C, using the relationship

$$[M=C^d \pmod n] \underline{M \equiv C^d \pmod n}$$

is used to develop the decryption sub-tasks:

$$[M_1 = C_1^{d_1} \bmod p_1] \underline{M_1 \equiv C_1^{d_1} \pmod {p_1}}$$

$$[M_2 = C_2^{d_2} \bmod p_2] \underline{M_2 \equiv C_2^{d_2} \pmod {p_2}}$$

$[M_3 = C_3^{d_3} \pmod{p_3}] \underline{M_3 \equiv C_3^{d_3} \pmod{p_3}}$

where

$[C_1 = C \pmod{p_1}] \underline{C_1 \equiv C \pmod{p_1}}$;

$[C_2 = C \pmod{p_2}] \underline{C_2 \equiv C \pmod{p_2}}$;

$[C_3 = C \pmod{p_3}] \underline{C_3 \equiv C \pmod{p_3}}$;

$[d_1 = d \pmod{(p_1 - 1)}] \underline{d_1 \equiv d \pmod{(p_1 - 1)}}$;

$[d_2 = d \pmod{(p_2 - 1)}] \underline{d_2 \equiv d \pmod{(p_2 - 1)}}$; and

$[d_3 = d \pmod{(p_3 - 1)}] \underline{d_3 \equiv d \pmod{(p_3 - 1)}}$.

Replace the paragraph beginning at col. 6, line 24 with the following:

The results of each sub-task, M_1 , M_2 , and M_3 can be combined to produce the plaintext, M , by a number of techniques. However, it is found that they can most expeditiously be combined by a form of the Chinese Remainder Theorem (CRT) using, preferably, a recursive scheme. Generally, the plaintext M is obtained from the combination of the individual sub-tasks by the following relationship:

$$Y_i \equiv Y_{i-1} + ((M_i - Y_{i-1}) (w_i^{-1} \pmod{p_i}) \pmod{p_i}) \cdot w_i \pmod{n} \quad [Y_i = Y_{i-1} + [(M_i - Y_{i-1}) (w_i^{-1} \pmod{p_i}) \pmod{p_i}] \cdot w_i \pmod{n}]$$

where $[i \geq 2] 2 \leq i \leq k$ where k is the number of prime factors of n , and

$$M = Y_k, \quad Y_1 = C_1, \quad \text{and} \quad w_i = \prod_{j < i} p_j$$

Encryption is performed in much the same manner as that used to obtain the plaintext M , provided (as noted above) the factors of n are available. Thus, the relationship

$[C = M^e \pmod{n}] \underline{C \equiv M^e \pmod{n}}$,

can be broken down into the three sub-tasks,

$[C_1 = M_1^{e_1} \bmod p_1]$ $C_1 = M_1^{e_1} \pmod{p_1}$,

$[C_2 = M_2^{e_2} \bmod p_2]$ $C_2 = M_2^{e_2} \pmod{p_2}$ and

$[C_3 = M_3^{e_3} \bmod p_3]$ $C_3 = M_3^{e_3} \pmod{p_3}$,

where

$[M_1 = M \pmod{p_1}]$ $M_1 \equiv M \pmod{p_1}$,

$[M_2 = M \pmod{p_2}]$ $M_2 \equiv M \pmod{p_2}$,

$[M_3 = M \pmod{p_3}]$ $M_3 \equiv M \pmod{p_3}$,

$[e_1 = e \pmod{(p_1 - 1)}]$ $e_1 \equiv e \pmod{p_1 - 1}$,

$[e_2 = e \pmod{(p_2 - 1)}]$ $e_2 \equiv e \pmod{p_2 - 1}$, and

$[e_3 = e \pmod{(p_3 - 1)}]$ $e_3 \equiv e \pmod{p_3 - 1}$.

Replace the paragraph beginning at col. 6, line 65 with the following:

In generalized form, the ciphertext C (i.e., [decrypted] encrypted message M) can be obtained by [the same summation] a recursive scheme as identified above to obtain the ciphertext C from its contiguous constituent sub-tasks C_i .

Replace the paragraph beginning at col. 7, line 1 with the following:

Preferably, the recursive CRT method described above is used to obtain either the ciphertext[,] C[,] or the deciphered plaintext (message) M due to its speed. However, there may be [occasions] implementations when it is beneficial to use a non-recursive technique in which case the following relationships are used:

$$M \equiv \sum_{i=1}^k M_i (w_i^{-1} \pmod{p_i}) \cdot w_i \pmod{n} \quad [M = \sum_{i=1}^k M_i (w_i^{-1} \bmod p_i) w_i \bmod n]$$

where

(2)
$$[w_i = \prod_{j \neq i} p_j] \underline{w_i = \prod_{j \neq i} p_j}, \text{ and}$$

k is the number (3 or more) of distinct primes chosen to develop the product *n*.

Replace the paragraph beginning at col. 7, line 17 with the following:

Thus, for example above (*k*=3), *M* is constructed from the returned sub-task values *M*₁, *M*₂, *M*₃ by the relationship

(2)
$$[M = M_1 (w_1^{-1} \bmod p_1) w_1 \bmod n + M_2 (w_2^{-1} \bmod p_2) w_2 \bmod n + M_3 (w_3^{-1} \bmod p_3) w_3 \bmod n] \underline{M \equiv M_1 (w_1^{-1} \bmod p_1) \cdot w_1 \bmod n} \\ + \underline{M_2 (w_2^{-1} \bmod p_2) \cdot w_2 \bmod n} \\ + \underline{M_3 (w_3^{-1} \bmod p_3) \cdot w_3 \bmod n}$$

where

$w_1 = p_2 p_3, w_2 = p_1 p_3, \text{ and } w_3 = p_1 p_2.$

Replace the paragraph beginning at col. 7, line 52 with the following:

The I/O bus 30 communicatively connects the CPU to a number of exponentiator elements [32_a, 32_b, and 32_c]. Shown here are three exponentiator elements, although as illustrated by the "other" exponentiators [32_n], additional exponentiator elements can be added. Each exponentiator element is a state machine controlled arithmetic circuit structured specifically to implement the relationship described above. Thus, for example, the exponentiator 32_a would be provided the values *M*₁, *e*₁, and *p*₁[, *n*] to develop *C*₁. Similarly, the exponentiator circuits 32_b and 32_c develop *C*₂ and *C*₃ from corresponding subtask values *M*₂, *e*₂, [P₂]p₂, *M*₃, *e*₃, and [P₃]p₃.

Replace the paragraph beginning at col. 8, line 1 with the following:

In order to ensure a secure environment, it is preferable that the cryptosystem 10 meet the Federal Information [Protection System] Processing Standard (FIPS) 140-1 level 3. Accordingly, the elements that make up the CPU 14 would be implemented in a design that will be secure from external probing of the circuit. However, information communicated on the I/O bus 30 between the CPU 14 and the exponentiator circuits 32 (and external memory 34--if present) is exposed. Consequently, to maintain the security of that information, it is first encrypted by the DES unit 24 before it is placed on the I/O bus 30 by the CPU 14. The exponentiator circuits 32, as well as the external memory 34, will also include similar DES units to decrypt information received from the CPU, and later to encrypt information returned to the CPU 14.

G24
G25
G26
G27
G28
G29
G30
G31
G32
G33
G34
G35
G36
G37
G38
G39
G40
G41
G42
G43
G44
G45
G46
G47
G48
G49
G50
G51
G52
G53
G54
G55
G56
G57
G58
G59
G60
G61
G62
G63
G64
G65
G66
G67
G68
G69
G70
G71
G72
G73
G74
G75
G76
G77
G78
G79
G80
G81
G82
G83
G84
G85
G86
G87
G88
G89
G90
G91
G92
G93
G94
G95
G96
G97
G98
G99
G100
G101
G102
G103
G104
G105
G106
G107
G108
G109
G110
G111
G112
G113
G114
G115
G116
G117
G118
G119
G120
G121
G122
G123
G124
G125
G126
G127
G128
G129
G130
G131
G132
G133
G134
G135
G136
G137
G138
G139
G140
G141
G142
G143
G144
G145
G146
G147
G148
G149
G150
G151
G152
G153
G154
G155
G156
G157
G158
G159
G160
G161
G162
G163
G164
G165
G166
G167
G168
G169
G170
G171
G172
G173
G174
G175
G176
G177
G178
G179
G180
G181
G182
G183
G184
G185
G186
G187
G188
G189
G190
G191
G192
G193
G194
G195
G196
G197
G198
G199
G200
G201
G202
G203
G204
G205
G206
G207
G208
G209
G210
G211
G212
G213
G214
G215
G216
G217
G218
G219
G220
G221
G222
G223
G224
G225
G226
G227
G228
G229
G230
G231
G232
G233
G234
G235
G236
G237
G238
G239
G240
G241
G242
G243
G244
G245
G246
G247
G248
G249
G250
G251
G252
G253
G254
G255
G256
G257
G258
G259
G260
G261
G262
G263
G264
G265
G266
G267
G268
G269
G270
G271
G272
G273
G274
G275
G276
G277
G278
G279
G280
G281
G282
G283
G284
G285
G286
G287
G288
G289
G290
G291
G292
G293
G294
G295
G296
G297
G298
G299
G300
G301
G302
G303
G304
G305
G306
G307
G308
G309
G310
G311
G312
G313
G314
G315
G316
G317
G318
G319
G320
G321
G322
G323
G324
G325
G326
G327
G328
G329
G330
G331
G332
G333
G334
G335
G336
G337
G338
G339
G340
G341
G342
G343
G344
G345
G346
G347
G348
G349
G350
G351
G352
G353
G354
G355
G356
G357
G358
G359
G360
G361
G362
G363
G364
G365
G366
G367
G368
G369
G370
G371
G372
G373
G374
G375
G376
G377
G378
G379
G380
G381
G382
G383
G384
G385
G386
G387
G388
G389
G390
G391
G392
G393
G394
G395
G396
G397
G398
G399
G400
G401
G402
G403
G404
G405
G406
G407
G408
G409
G410
G411
G412
G413
G414
G415
G416
G417
G418
G419
G420
G421
G422
G423
G424
G425
G426
G427
G428
G429
G430
G431
G432
G433
G434
G435
G436
G437
G438
G439
G440
G441
G442
G443
G444
G445
G446
G447
G448
G449
G450
G451
G452
G453
G454
G455
G456
G457
G458
G459
G460
G461
G462
G463
G464
G465
G466
G467
G468
G469
G470
G471
G472
G473
G474
G475
G476
G477
G478
G479
G480
G481
G482
G483
G484
G485
G486
G487
G488
G489
G490
G491
G492
G493
G494
G495
G496
G497
G498
G499
G500
G501
G502
G503
G504
G505
G506
G507
G508
G509
G510
G511
G512
G513
G514
G515
G516
G517
G518
G519
G520
G521
G522
G523
G524
G525
G526
G527
G528
G529
G530
G531
G532
G533
G534
G535
G536
G537
G538
G539
G540
G541
G542
G543
G544
G545
G546
G547
G548
G549
G550
G551
G552
G553
G554
G555
G556
G557
G558
G559
G560
G561
G562
G563
G564
G565
G566
G567
G568
G569
G570
G571
G572
G573
G574
G575
G576
G577
G578
G579
G580
G581
G582
G583
G584
G585
G586
G587
G588
G589
G590
G591
G592
G593
G594
G595
G596
G597
G598
G599
G600
G601
G602
G603
G604
G605
G606
G607
G608
G609
G610
G611
G612
G613
G614
G615
G616
G617
G618
G619
G620
G621
G622
G623
G624
G625
G626
G627
G628
G629
G630
G631
G632
G633
G634
G635
G636
G637
G638
G639
G640
G641
G642
G643
G644
G645
G646
G647
G648
G649
G650
G651
G652
G653
G654
G655
G656
G657
G658
G659
G660
G661
G662
G663
G664
G665
G666
G667
G668
G669
G670
G671
G672
G673
G674
G675
G676
G677
G678
G679
G680
G681
G682
G683
G684
G685
G686
G687
G688
G689
G690
G691
G692
G693
G694
G695
G696
G697
G698
G699
G700
G701
G702
G703
G704
G705
G706
G707
G708
G709
G710
G711
G712
G713
G714
G715
G716
G717
G718
G719
G720
G721
G722
G723
G724
G725
G726
G727
G728
G729
G730
G731
G732
G733
G734
G735
G736
G737
G738
G739
G740
G741
G742
G743
G744
G745
G746
G747
G748
G749
G750
G751
G752
G753
G754
G755
G756
G757
G758
G759
G760
G761
G762
G763
G764
G765
G766
G767
G768
G769
G770
G771
G772
G773
G774
G775
G776
G777
G778
G779
G780
G781
G782
G783
G784
G785
G786
G787
G788
G789
G790
G791
G792
G793
G794
G795
G796
G797
G798
G799
G800
G801
G802
G803
G804
G805
G806
G807
G808
G809
G810
G811
G812
G813
G814
G815
G816
G817
G818
G819
G820
G821
G822
G823
G824
G825
G826
G827
G828
G829
G830
G831
G832
G833
G834
G835
G836
G837
G838
G839
G840
G841
G842
G843
G844
G845
G846
G847
G848
G849
G850
G851
G852
G853
G854
G855
G856
G857
G858
G859
G860
G861
G862
G863
G864
G865
G866
G867
G868
G869
G870
G871
G872
G873
G874
G875
G876
G877
G878
G879
G880
G881
G882
G883
G884
G885
G886
G887
G888
G889
G890
G891
G892
G893
G894
G895
G896
G897
G898
G899
G900
G901
G902
G903
G904
G905
G906
G907
G908
G909
G910
G911
G912
G913
G914
G915
G916
G917
G918
G919
G920
G921
G922
G923
G924
G925
G926
G927
G928
G929
G930
G931
G932
G933
G934
G935
G936
G937
G938
G939
G940
G941
G942
G943
G944
G945
G946
G947
G948
G949
G950
G951
G952
G953
G954
G955
G956
G957
G958
G959
G960
G961
G962
G963
G964
G965
G966
G967
G968
G969
G970
G971
G972
G973
G974
G975
G976
G977
G978
G979
G980
G981
G982
G983
G984
G985
G986
G987
G988
G989
G990
G991
G992
G993
G994
G995
G996
G997
G998
G999
G1000
G1001
G1002
G1003
G1004
G1005
G1006
G1007
G1008
G1009
G1010
G1011
G1012
G1013
G1014
G1015
G1016
G1017
G1018
G1019
G1020
G1021
G1022
G1023
G1024
G1025
G1026
G1027
G1028
G1029
G1030
G1031
G1032
G1033
G1034
G1035
G1036
G1037
G1038
G1039
G1040
G1041
G1042
G1043
G1044
G1045
G1046
G1047
G1048
G1049
G1050
G1051
G1052
G1053
G1054
G1055
G1056
G1057
G1058
G1059
G1060
G1061
G1062
G1063
G1064
G1065
G1066
G1067
G1068
G1069
G1070
G1071
G1072
G1073
G1074
G1075
G1076
G1077
G1078
G1079
G1080
G1081
G1082
G1083
G1084
G1085
G1086
G1087
G1088
G1089
G1090
G1091
G1092
G1093
G1094
G1095
G1096
G1097
G1098
G1099
G1100
G1101
G1102
G1103
G1104
G1105
G1106
G1107
G1108
G1109
G1110
G1111
G1112
G1113
G1114
G1115
G1116
G1117
G1118
G1119
G1120
G1121
G1122
G1123
G1124
G1125
G1126
<

A2
[n] p_1 , to the exponentiator 32a. Similar values will be developed by the processor 20 for the sub-tasks that will be delivered to the exponentiators 32b and 32c.

A2
Replace the paragraph beginning at col. 10, line 15 with the following:

A2
Alternatively, the [post]host-system 50 may desire to deliver, via the communication medium 60, an encrypted communication to one of the stations 64. If the communication is to be encrypted by the DES scheme, with the DES key encrypted by the RSA scheme, the host system would encrypt the communication, forward the DES key to one of the cryptosystems 10 for encryption via the RSA scheme. When the encrypted DES key is received back from the cryptosystem 10, the host system can then deliver to one or more of the stations 64 the encrypted message.

A2
Replace the paragraph beginning at col. 10, line 25 with the following:

A2
Of course, the host system 50 and the stations 64 will be using the RSA scheme of public key encryption/decryption. Encrypted communications from the stations 64 to the host system 50 require that the stations 64 have access to the public key $[E (E, N)]$ $E=(e, n)$ while the host system maintains the private key $[D (D, N)]$ $D=(d, n)$ and the constituent primes, p_1, p_2, \dots, p_k . Conversely, for secure communication from the host system 50 to one or more of the stations 64, the host system would retain a public key E' for each station 64, while the stations retain the corresponding private keys $[E']$ D' .

A2
Replace the paragraph beginning at col. 10, line 35 with the following:

A2
Other techniques for encrypting the communication could be used. For example, the communication could be entirely encrypted by the RSA scheme. If, however, the message to be communicated [on] is represented by a numerical value greater than $n-1$, it will need to be broken up into blocks size M where

$$[0 \leq M \leq N-1] \quad 0 \leq M \leq n-1.$$